

## Ruby trunk - Bug #9452

### Refining methods that should be private

01/26/2014 03:01 PM - rbjl (Jan Lejis)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	shugo (Shugo Maeda)	
<b>Target version:</b>		
<b>ruby -v:</b>	ruby 2.2.0dev (2014-01-25 trunk 44707) [x86_64-linux]	<b>Backport:</b> 1.9.3: DONTNEED, 2.0.0: DONE, 2.1: DONE

#### Description

Are refinements also meant to add private methods? This is what I tried:

```
module R
  refine Object do
    def m
      puts "Success!"
    end
  end
end
```

```
private(:m)
end
```

```
end
```

```
using R
```

```
m # success
```

```
42.m # success (= not private)
```

However, I can get near the desired functionality by defining a private method first:

```
class Object
  private
  def m
    end
end
```

```
module R
  refine Object do
    def m
      puts "Success!"
    end
  end
end
```

```
end
```

```
using R
```

```
m # success
```

```
42.m # no success (= it is private)
```

It calls the right code. But requires global core ext.

#### Associated revisions

Revision 4d33c0e9 - 02/13/2014 02:44 PM - shugo (Shugo Maeda)

- `vm_inshelper.c` (`vm_call_method`): should check `ci->me->flag` of a refining method in case the method is private. [ruby-core:60111] [Bug #9452]
- `vm_method.c` (`make_method_entry_refined`): set `me->flag` of a refined method entry to `NOEX_PUBLIC` in case the original method is private and it is refined as a public method. The original flag is stored

in me->def->body.orig\_me, so it's OK to make a refined method entry public. [ruby-core:60111] [Bug #9452]

- test/ruby/test\_refinement.rb: related tests.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@44931 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### Revision 44931 - 02/13/2014 02:44 PM - shugo (Shugo Maeda)

- vm\_inshelper.c (vm\_call\_method): should check ci->me->flag of a refining method in case the method is private. [ruby-core:60111] [Bug #9452]
- vm\_method.c (make\_method\_entry\_refined): set me->flag of a refined method entry to NOEX\_PUBLIC in case the original method is private and it is refined as a public method. The original flag is stored in me->def->body.orig\_me, so it's OK to make a refined method entry public. [ruby-core:60111] [Bug #9452]
- test/ruby/test\_refinement.rb: related tests.

#### Revision 44931 - 02/13/2014 02:44 PM - shugo (Shugo Maeda)

- vm\_inshelper.c (vm\_call\_method): should check ci->me->flag of a refining method in case the method is private. [ruby-core:60111] [Bug #9452]
- vm\_method.c (make\_method\_entry\_refined): set me->flag of a refined method entry to NOEX\_PUBLIC in case the original method is private and it is refined as a public method. The original flag is stored in me->def->body.orig\_me, so it's OK to make a refined method entry public. [ruby-core:60111] [Bug #9452]
- test/ruby/test\_refinement.rb: related tests.

#### Revision 44931 - 02/13/2014 02:44 PM - shugo (Shugo Maeda)

- vm\_inshelper.c (vm\_call\_method): should check ci->me->flag of a refining method in case the method is private. [ruby-core:60111] [Bug #9452]
- vm\_method.c (make\_method\_entry\_refined): set me->flag of a refined method entry to NOEX\_PUBLIC in case the original method is private and it is refined as a public method. The original flag is stored in me->def->body.orig\_me, so it's OK to make a refined method entry public. [ruby-core:60111] [Bug #9452]
- test/ruby/test\_refinement.rb: related tests.

#### Revision 44931 - 02/13/2014 02:44 PM - shugo (Shugo Maeda)

- vm\_inshelper.c (vm\_call\_method): should check ci->me->flag of a refining method in case the method is private. [ruby-core:60111] [Bug #9452]
- vm\_method.c (make\_method\_entry\_refined): set me->flag of a refined

method entry to NOEX\_PUBLIC in case the original method is private and it is refined as a public method. The original flag is stored in me->def->body.orig\_me, so it's OK to make a refined method entry public. [ruby-core:60111] [Bug #9452]

- test/ruby/test\_refinement.rb: related tests.

#### Revision 44931 - 02/13/2014 02:44 PM - shugo (Shugo Maeda)

- vm\_inshelper.c (vm\_call\_method): should check ci->me->flag of a refining method in case the method is private. [ruby-core:60111] [Bug #9452]
- vm\_method.c (make\_method\_entry\_refined): set me->flag of a refined method entry to NOEX\_PUBLIC in case the original method is private and it is refined as a public method. The original flag is stored in me->def->body.orig\_me, so it's OK to make a refined method entry public. [ruby-core:60111] [Bug #9452]
- test/ruby/test\_refinement.rb: related tests.

#### Revision 44931 - 02/13/2014 02:44 PM - shugo (Shugo Maeda)

- vm\_inshelper.c (vm\_call\_method): should check ci->me->flag of a refining method in case the method is private. [ruby-core:60111] [Bug #9452]
- vm\_method.c (make\_method\_entry\_refined): set me->flag of a refined method entry to NOEX\_PUBLIC in case the original method is private and it is refined as a public method. The original flag is stored in me->def->body.orig\_me, so it's OK to make a refined method entry public. [ruby-core:60111] [Bug #9452]
- test/ruby/test\_refinement.rb: related tests.

#### Revision ab530f05 - 02/15/2014 06:53 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) r44931: [Backport #9452]

- \* vm\_inshelper.c (vm\_call\_method): should check ci->me->flag of a refining method in case the method is private. [ruby-core:60111] [Bug #9452]
- \* vm\_method.c (make\_method\_entry\_refined): set me->flag of a refined method entry to NOEX\_PUBLIC in case the original method is private and it is refined as a public method. The original flag is stored in me->def->body.orig\_me, so it's OK to make a refined method entry public. [ruby-core:60111] [Bug #9452]
- \* test/ruby/test\_refinement.rb: related tests.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_2\_0\_0@44967 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### Revision 669e9ac5 - 02/22/2014 05:05 AM - naruse (Yui NARUSE)

merge revision(s) 44931: [Backport #9452]

- \* vm\_inshelper.c (vm\_call\_method): should check ci->me->flag of a refining method in case the method is private. [ruby-core:60111] [Bug #9452]
- \* vm\_method.c (make\_method\_entry\_refined): set me->flag of a refined method entry to NOEX\_PUBLIC in case the original method is private

```
and it is refined as a public method. The original flag is stored
in me->def->body.orig_me, so it's OK to make a refined method
entry public. [ruby-core:60111] [Bug #9452]
```

\* test/ruby/test\_refinement.rb: related tests.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_2\_1@45107 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

## History

---

### #1 - 02/13/2014 08:21 AM - shugo (Shugo Maeda)

- Assignee set to shugo (Shugo Maeda)

### #2 - 02/13/2014 02:44 PM - shugo (Shugo Maeda)

- Status changed from Open to Closed

- % Done changed from 0 to 100

Applied in changeset [r44931](#).

---

- vm\_inshelper.c (vm\_call\_method): should check ci->me->flag of a refining method in case the method is private. [ruby-core:60111] [Bug [#9452](#)]
- vm\_method.c (make\_method\_entry\_refined): set me->flag of a refined method entry to NOEX\_PUBLIC in case the original method is private and it is refined as a public method. The original flag is stored in me->def->body.orig\_me, so it's OK to make a refined method entry public. [ruby-core:60111] [Bug [#9452](#)]
- test/ruby/test\_refinement.rb: related tests.

### #3 - 02/14/2014 04:18 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 1.9.3: UNKNOWN, 2.0.0: UNKNOWN, 2.1: UNKNOWN to 1.9.3: DONTNEED, 2.0.0: REQUIRED, 2.1: REQUIRED

### #4 - 02/15/2014 06:53 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 1.9.3: DONTNEED, 2.0.0: REQUIRED, 2.1: REQUIRED to 1.9.3: DONTNEED, 2.0.0: DONE, 2.1: REQUIRED

[r44931](#) was backported to ruby\_2\_0\_0 at r44967.

### #5 - 02/22/2014 05:05 AM - naruse (Yui NARUSE)

- Backport changed from 1.9.3: DONTNEED, 2.0.0: DONE, 2.1: REQUIRED to 1.9.3: DONTNEED, 2.0.0: DONE, 2.1: DONE

r45107.